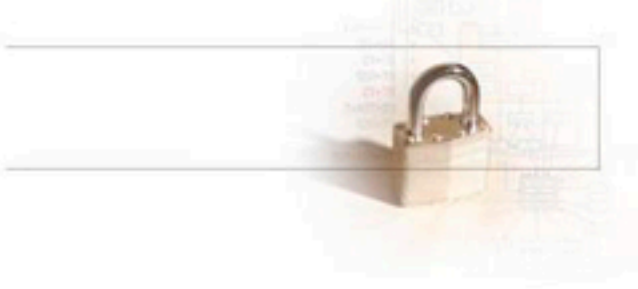


The Security Development Lifecycle (SDL)

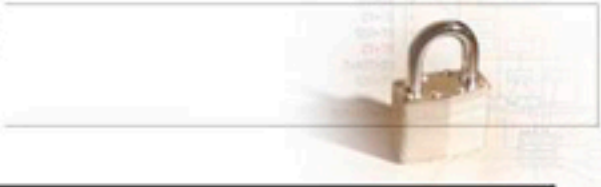
Security Engineering & Communications

Overview

- Security Development Lifecycle (SDL) formalizes security best practices from Windows Server 2003, .NET Framework v1, other security pushes of late 2001, early 2002
  - Development process
    - Threat modeling
    - Security Push
    - Final Security Review (FSR) – formerly security audit
  - Security education
- SDL is a best practice mandated by SteveB & SLT. Properly implemented, it will produce the best more secure products possible for our customers
  - Process
  - Education
  - Accountability

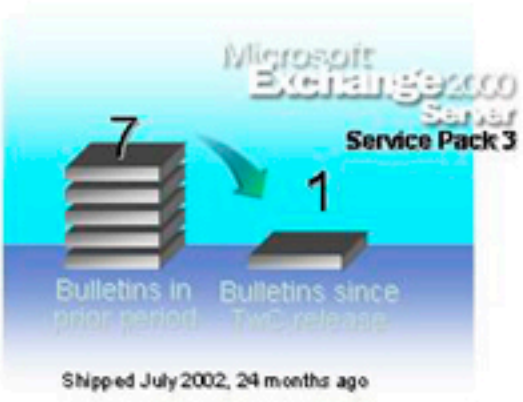
Process	Education	Accountability
• Defined milestones tied to product lifecycle • New resources – SWI buddy – Guidance and tools • New deliverables – Early threat models – Fuzz testing	• Annual engineer training • Tracking of training compliance • On-line training options • Specialized training options	• Business group CPE metrics by product • Training metrics by organization

Security Engineering Background



How we got here	• Security push or stand down originated with .NET Framework 1.0 and Windows Server 2003 • Final Security Review (FSR) by SWI team initiated before Windows Server 2003 RTM. FSR proves effective. • Security push and FSR extended to other products
What product teams have done	• Develop threat models to focus security analysis, code review, attack testing on the high risk areas • Follow guidelines on coding, testing, tool usage • Conduct a security push at code complete (after Beta 1)
What SWI team has done (key components)	• Maintain a web repository of tools and guidance (http://swi) • Answer questions on security development practices • Conduct FSR • Investigate reported security vulnerabilities and initiate "security response lockdown" for systemic/major issues
What we've achieved	• Over fifty security pushes and forty FSRs across Microsoft; over 100 total engagements including consulting, reviews, training; over 100 patch-class issues found during FSRs • Over 10,000 engineers trained across Microsoft (most during Windows Server 2003 security push) • Products that pushed through the process are measurably more secure

Impact of Security Engineering on Products

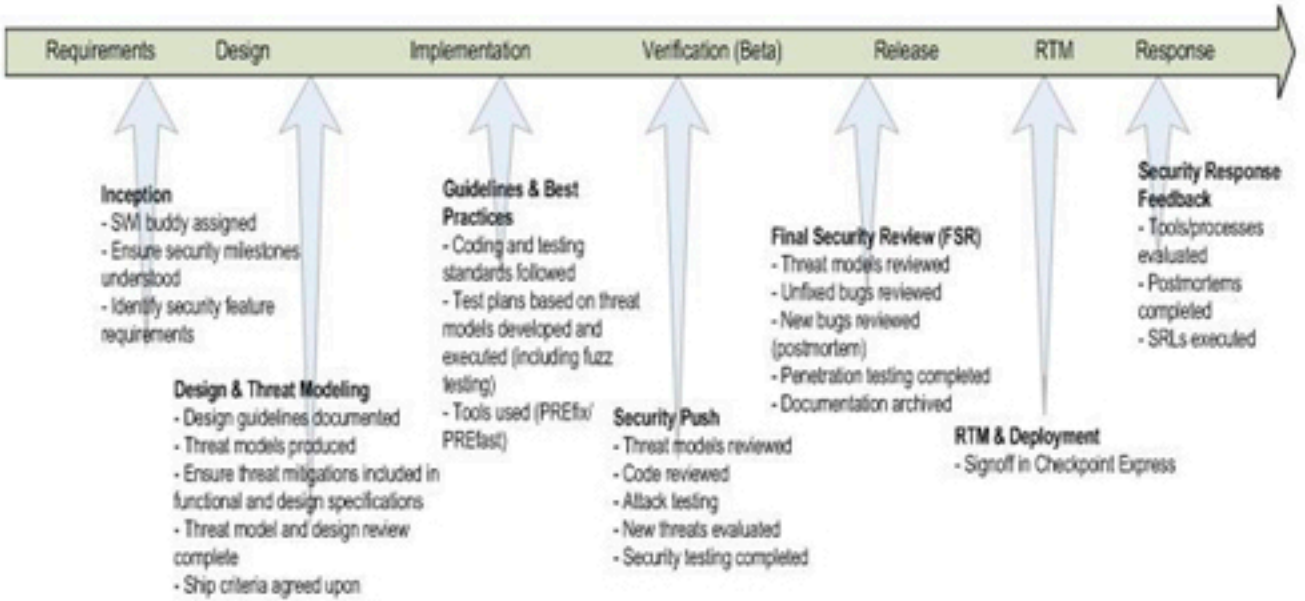


Critical or Important vulnerabilities in the first…

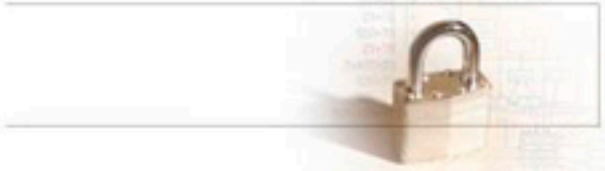
	…270 days	…365 days	TwC release?
Windows 2000 Servers	36	42	No
Windows Server 2003	6	13	Yes

Critically important to review "old code"

SDL Process

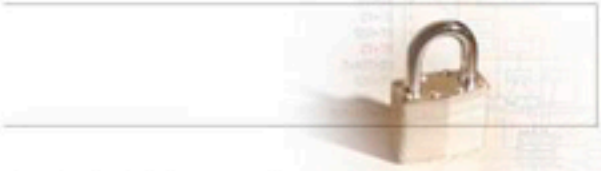

Requirements
Design
Implementation
Verification (Beta)
Release
RTM
Response
Inception
- SM buddy assigned
- Ensure security milestones understood
- Identify security feature requirements
Design & Threat Modeling
- Design guidelines documented
- Threat models produced
- Ensure threat mitigations included in functional and design specifications
- Threat model and design review complete
- Ship criteria agreed upon
Guidelines & Best Practices
- Coding and testing standards followed
- Test plans based on threat models developed and executed (including fuzz testing)
- Tools used (PREfix/ PREfast)
Security Push
- Threat models reviewed
- Code reviewed
- Attack testing
- New threats evaluated
- Security testing completed
Final Security Review (FSR)
- Threat models reviewed
- Unfixed bugs reviewed
- New bugs reviewed (postmortem)
- Penetration testing completed
- Documentation archived
RTM & Deployment
- Signoff in Checkpoint Express
Security Response Feedback
- Tools/processes evaluated
- Postmortems completed
- SRLs executed

SDL Education



Education Focus	**Mandatory training & education:**	
	• Mandatory 4-hour training for all engineers – Applies to Dev, Test, PM, UA, UE assigned to products where SDL is mandatory – Required if previous training was prior to 1 October 2003 – Will be offered at major remote development sites	• Targeted for completion by 7/04
	• On-line training option under development – Appropriate for remote sites – May be alternative to live training	• In final editing now
	• Specialized course offerings will satisfy mandatory training requirement	• In planning – to roll out in FY05
Accountability for learning	**Track and evaluate engineer training**	
	• Record all engineer training	• Now
	• Develop improved training transcript system	• Begins FY05
	• Develop employee training assessment system	• Begins FY05
	• Reflect training level (classes, assessment) in reviews, promotion, assignments	• Begins FY05

SDL Accountability



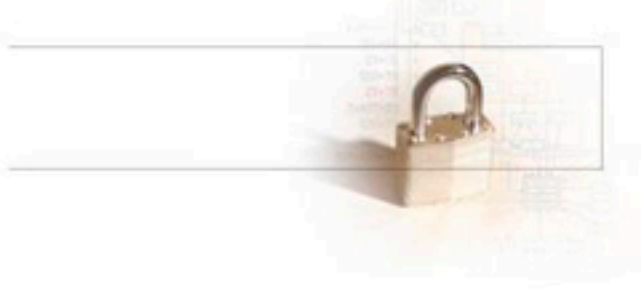
Product Team	Training Participation	Product Version	SDL Compliant	Threat Models Complete at Design	Pri1 Code Reviewed	Percent Threats Covered by Tests	Critical or Important Vulnerabilities at FSR	Critical or Important Vulnerabilities in Field	Total Vulnerabilities in Field	Security Update Problems
Windows Server	%	2003	Y	%	%	%	Number	Number	Number	Number
		2000	N	NA	NA	NA	NA	Number	Number	Number
Exchange	%	2003	Y	%	%	%	Number	Number	Number	Number
		2000	Y	NA	NA	NA	NA	Number	Number	Number
SQL Server	%	Yukon	Y	%	%	%	Number	Number	Number	Number
		2000	Y	NA	NA	NA	NA	Number	Number	Number
ISA Server	%	2004	Y	%	%	%	Number	Number	Number	Number
		2000	N	NA	NA	NA	NA	Number	Number	Number
SMS	%	2003	Y	%	%	%	Number	Number	Number	Number
		2000	N	NA	NA	NA	NA	Number	Number	Number
other	%									
other	%									
other	%									
	100%			100%	100%	>90%	0	Per Product	Per Product	0
	>90%			>80%	>90%	>75%	<=2	Per Product	Per Product	< 10%
	<90%			<80%	<90%	<75%	>2	Per Product	Per Product	> 10%

On Track
Attention Required
Critical Issue

- CPE metrics reviewed beginning late CY 04
- Training metrics tracked by team
- Other metrics tracked by product

Planned Additions to SDL

- Change process
  - SDL process documents updated as needed
  - SDL requirements updated twice yearly
  - New requirements posted as "recommendations" three months before effective date
- Current plans
  - Training updates and accountability
  - Security architecture and Trusted Computing Base
  - Defining and measuring attack surface
  - Privacy requirements
  - Integrated process as alternative to security push



Demonstration/Questions